Introduction

In this project, I will analyze the spread of the new corona virus. I will use this dataset: - The John Hopkins University's dataset which contains aggregated daily data for confirmed cases, deaths and recovered patients. https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series (https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series)

In [15]:
```
%matplotlib inline

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [16]:
```
url_case = 'https://github.com/CSSEGISandData/COVID-19/raw/master/csse
url_death = 'https://github.com/CSSEGISandData/COVID-19/raw/master/css
url_cured = 'https://github.com/CSSEGISandData/COVID-19/raw/master/css
```

Create a function to transform all 3 datasets

In [17]:
```
def load(url, measure_name):
    df = pd.read_csv(url, index_col=[0, 1, 2, 3])
    df = df.stack()
    df = df.reset_index()

    cum = 'cum_' + measure_name
    df.columns = ['prov_state', 'country', 'lat', 'long', 'date', cum]

    df.date = pd.to_datetime(df.date, format='%m/%d/%y')

    df = df[df[cum] != 0]

    df['location'] = np.where(df.prov_state.isnull(), df.country, df.p

    if measure_name == 'case':
        df = error_correction(df)

    new = 'new_' + measure_name
    df[new] = df.groupby('location')[cum].diff(1)
    df[new] = df[new].fillna(df[cum])

    return df[['location', 'prov_state', 'country', 'lat', 'long', 'da


def error_correction(df):
    df.loc[(df.location == 'Japan') & (df.date == '2020-03-12'), 'cum_
    df.loc[(df.location == 'Italy') & (df.date == '2020-03-12'), 'cum_
    df.loc[(df.location == 'Spain') & (df.date == '2020-03-12'), 'cum_
    df.loc[(df.location == 'Switzerland') & (df.date == '2020-03-12'),
    df.loc[(df.location == 'Netherlands') & (df.date == '2020-03-12'),
    df.loc[(df.location == 'France') & (df.date == '2020-03-12'), 'cum

    return df
```

In [18]:
```
case = load(url_case, 'case')
death = load(url_death, 'death')
cured = load(url_cured, 'cured')
```

In [19]:
```
case.shape
```

Out[19]: (116314, 8)

In [20]:
```
death.shape
```

Out[20]: (97439, 8)

In [21]:
```
cured.shape
```

Out[21]: (104555, 8)

In [22]:
```
# Merge into 1 dataframe.
df = pd.merge(case, death, how='left', on=['location', 'prov_state', '
df = pd.merge(df, cured, how='left', on=['location', 'prov_state', 'co
```

In [23]:
```
df = df.set_index('date')
```

In [24]:
```
df = df.replace({'Martinique':'France',
                 'Reunion':'France',
                 'French Guiana':'France',
                 'Guadeloupe': 'France',
                 'Mayotte': 'France',
                 'Aruba': 'Netherlands',
                 'Curacao': 'Netherlands',
                 'Guernsey':'United Kingdom',
                 'Jersey':'United Kingdom',
                 'Guam': 'US'})
```

In [25]:
```
df.head()
```

Out[25]:

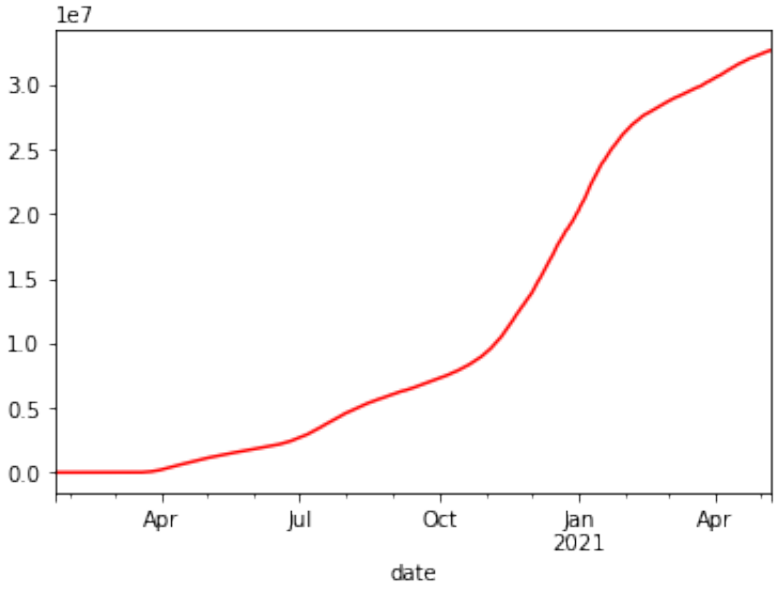
date	location	prov_state	country	lat	long	cum_case	new_case	cum_de
2020-02-24	Afghanistan	NaN	Afghanistan	33.93911	67.709953	1	1.0	N
2020-02-25	Afghanistan	NaN	Afghanistan	33.93911	67.709953	1	0.0	N
2020-02-26	Afghanistan	NaN	Afghanistan	33.93911	67.709953	1	0.0	N
2020-02-27	Afghanistan	NaN	Afghanistan	33.93911	67.709953	1	0.0	N
2020-02-28	Afghanistan	NaN	Afghanistan	33.93911	67.709953	1	0.0	N

Analysis

Cumulative cases in China

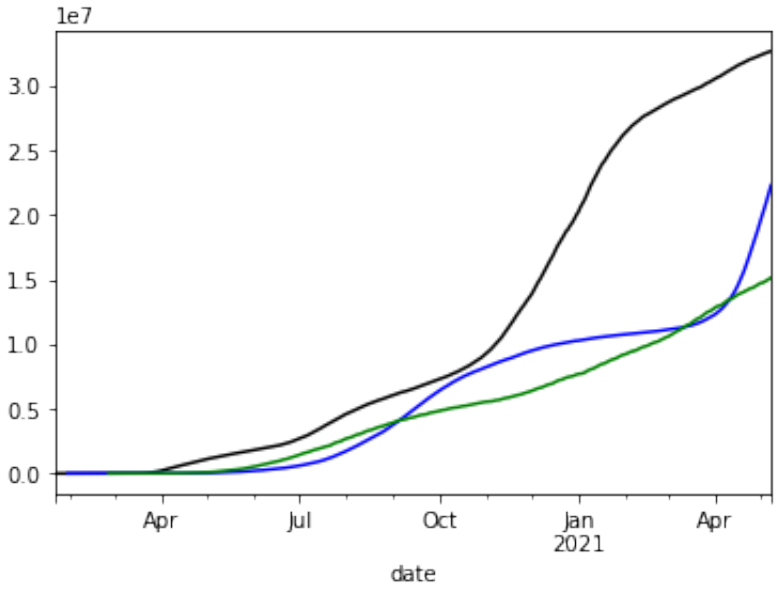
In [26]:
```
df[df.country == 'US'].groupby('date').sum().cum_case.plot(kind='line'
```

Out[26]: <AxesSubplot:xlabel='date'>



In [27]:
```
df[df.country == 'US'].groupby('date').sum().cum_case.plot(kind='line'
df[df.country == 'India'].groupby('date').sum().cum_case.plot(kind='li
df[df.country == 'Brazil'].groupby('date').sum().cum_case.plot(kind='l
```

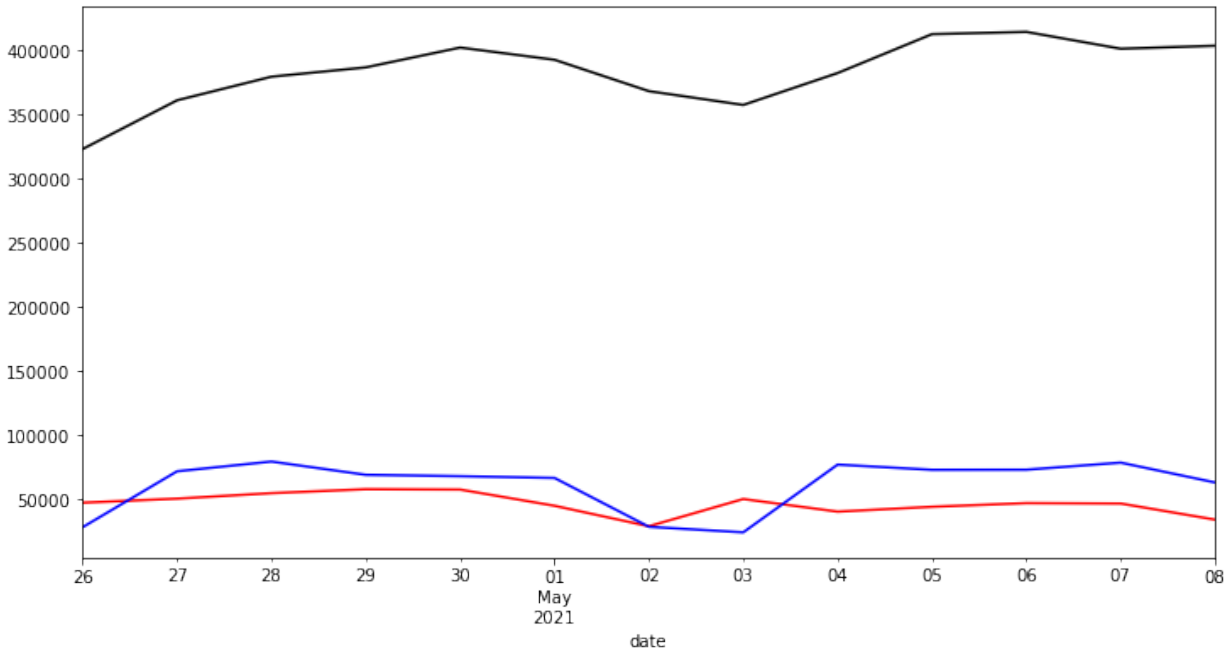
Out[27]: <AxesSubplot:xlabel='date'>



New cases by selected countries

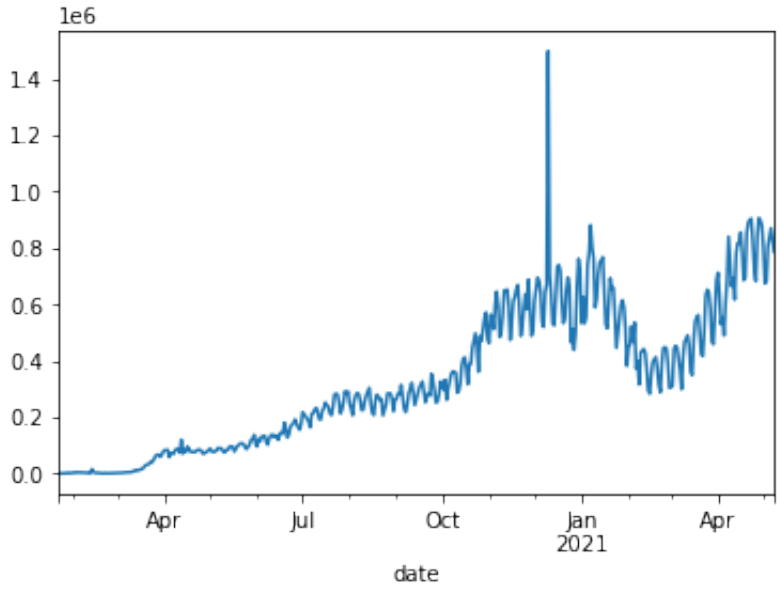
In [14]:
```
df_last_2_weeks = df.sort_index().last('2W')
```

In [28]:
```
igure(figsize=(12,6))
df_last_2_weeks[df_last_2_weeks.country == 'US'].resample('D').sum().n
 = df_last_2_weeks[df_last_2_weeks.country == 'India'].resample('D').s
l = df_last_2_weeks[df_last_2_weeks.country == 'Brazil'].resample('D')
```



In [29]:
```
df.groupby('date').new_case.sum().plot(kind='line')
```

Out[29]: <AxesSubplot:xlabel='date'>



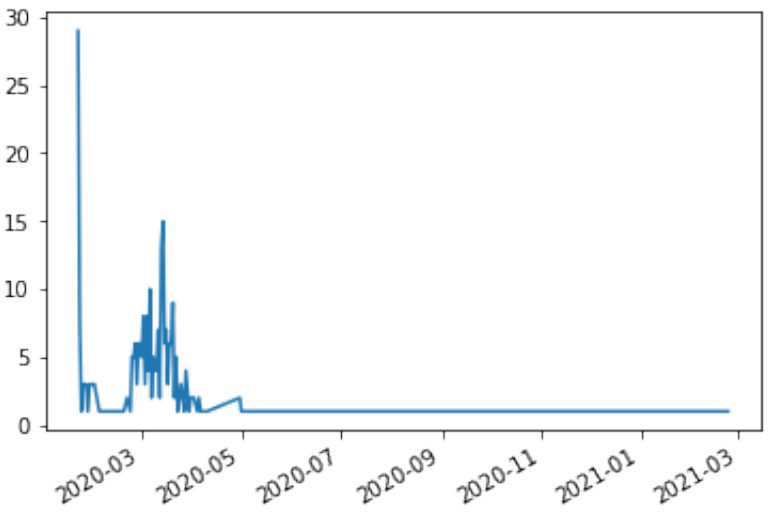
Number of new locations reporting first case, by date

In [30]:
```
first_case = df[df.cum_case.notnull()].reset_index().groupby(['locatic
```

In [31]:
```
first_case_by_date = first_case.date.value_counts()
```

In [32]:
```
first_case_by_date.sort_index().plot(kind='line')
```

Out[32]: <AxesSubplot:>



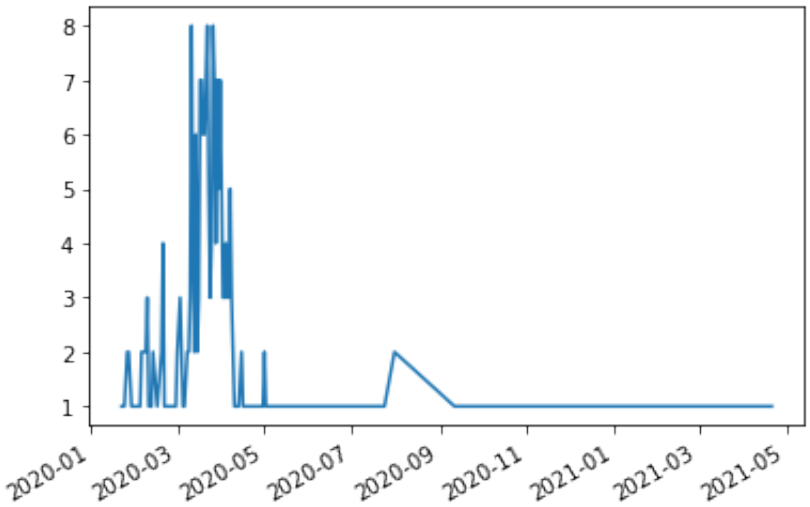
Number of new locations reporting death, by date

In [33]:
```
first_death = df[df.cum_death.notnull()].reset_index().groupby(['locat
```

In [34]:
```
first_death_by_date = first_death.date.value_counts()
```

In [35]:
```
first_death_by_date.sort_index().plot(kind='line')
```

Out[35]: <AxesSubplot:>

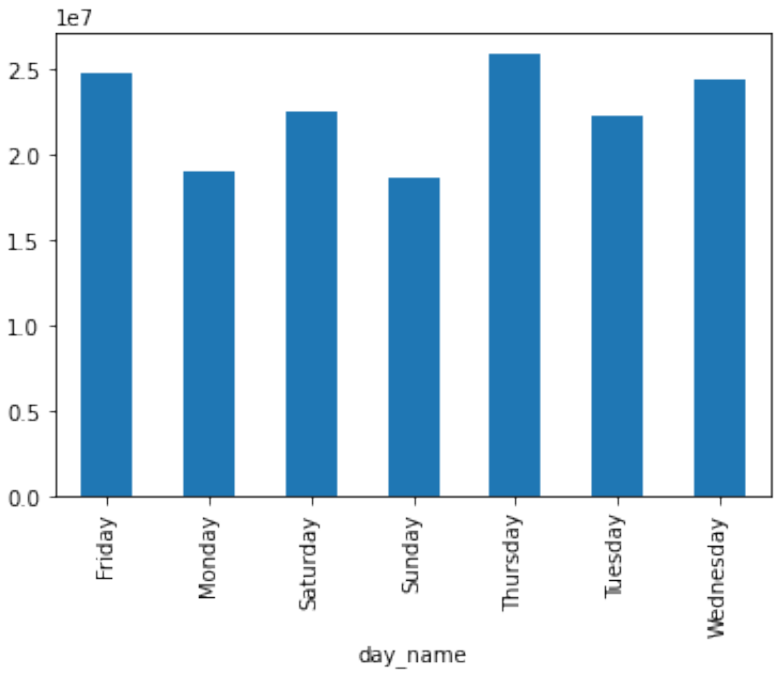


Playing with datetime index

In [36]:
```
df['day_name'] = df.index.day_name()
```

In [37]:
```
df.groupby('day_name').sum().new_case.plot(kind='bar')
```

Out[37]:
```
<AxesSubplot:xlabel='day_name'>
```



In [38]:
```
# Weekly new cases
df.resample('W').sum().new_case
```

Out[38]:
```
date
2020-01-26       2118.0
2020-02-02      14669.0
2020-02-09      23372.0
2020-02-16      31075.0
2020-02-23       7747.0
                 ...
2021-04-11    4721160.0
2021-04-18    5358370.0
2021-04-25    5784620.0
2021-05-02    5680889.0
2021-05-09    4817719.0
Freq: W-SUN, Name: new_case, Length: 68, dtype: float64
```

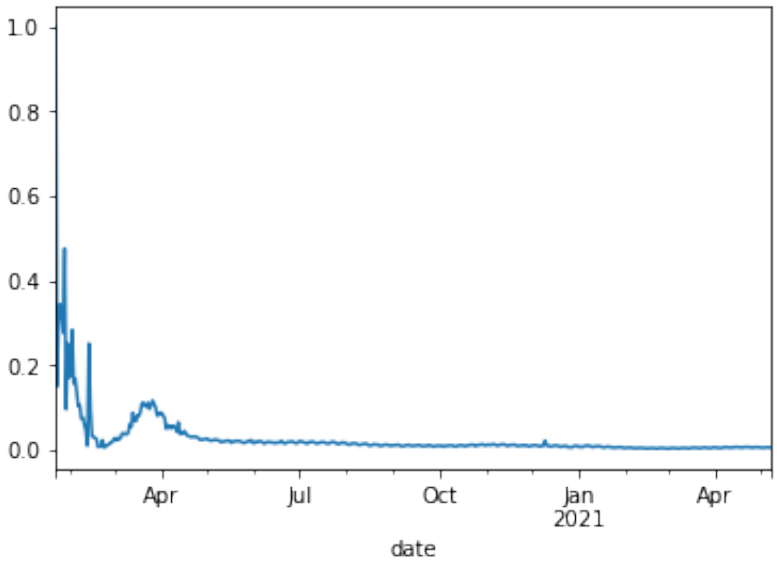
In [39]:
```
# Weekly new cases, excluding China
df[df.country != 'China'].resample('W').sum().new_case
```

Out[39]:
```
date
2020-01-26         43.0
2020-02-02        114.0
2020-02-09        173.0
2020-02-16        391.0
2020-02-23       1238.0
                  ...
2021-04-11    4720965.0
2021-04-18    5358184.0
2021-04-25    5784478.0
2021-05-02    5680743.0
2021-05-09    4817636.0
Freq: W-SUN, Name: new_case, Length: 68, dtype: float64
```

In [40]:
```
gbdate = df.groupby('date').sum()
```

In [41]:
```
(gbdate.new_case/gbdate.cum_case).plot()
```

Out[41]:
```
<AxesSubplot:xlabel='date'>
```



In [ ]:
```
```